Power Grid Defense Against Malicious Cascading Failure

Paulo Shakarian
Dept. EECS and
Network Science Center
U.S. Military Academy
West Point, NY, 10996
paulo[at]shakarian.net

Hansheng Lei
Dept. EECS and
Network Science Center
U.S. Military Academy
West Point, NY, 10996
hansheng.lei[at]usma.edu

Roy Lindelauf
Netherlands Defence
Academy
Faculty of Military Science
Military Operational Art and
Science
rha.lindelauf.01[at]nlda.nl

ABSTRACT

An adversary looking to disrupt a power grid may look to target certain substations and sources of power generation to initiate a cascading failure that maximizes the number of customers without electricity. This is particularly an important concern when the enemy has the capability to launch cyber-attacks as practical concerns (i.e. avoiding disruption of service, presence of legacy systems, etc.) may hinder security. Hence, a defender can harden the security posture at certain power stations but may lack the time and resources to do this for the entire power grid. We model a power grid as a graph and introduce the cascading failure game in which both the defender and attacker choose a subset of power stations such as to minimize (maximize) the number of consumers having access to producers of power. We formalize problems for identifying both mixed and deterministic strategies for both players, prove complexity results under a variety of different scenarios, identify tractable cases, and develop algorithms for these problems. We also perform an experimental evaluation of the model and game on a real-world power grid network. Empirically, we noted that the game favors the attacker as he benefits more from increased resources than the defender. Further, the minimax defense produces roughly the same expected payoff as an easy-to-compute deterministic load based (DLB) defense when played against a minimax attack strategy. However, DLB performs more poorly than minimax defense when faced with the attacker's best response to DLB. This is likely due to the presence of low-load yet high-payoff nodes, which we also found in our empirical analysis.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Algorithms Security

Keywords

power grid defense, game theory, complex networks

Appears in: *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), Lomuscio, Scerri, Bazzan, Huhns (eds.), May, 5–9, 2014, Paris, France.*

1. INTRODUCTION

Rapid cascading failure in a power grid caused by a succession of overloading lines can lead to very large outages, as observed in the United States in 2003 [1]. Studies on cascading failure [7, 8, 16] have illustrated that such a failure can be initiated with only a small number of initial node failures. Further, power grid infrastructure is often particularly vulnerable with respect to cyber-security due to a variety of issues, including the use of legacy and proprietary computer hardware and software [26].

In this paper, we extend the work on cascading failure models to a two-player game where an attacker attempts to create a cascade that maximizes the number of customers without power while the defender defends key nodes to avoid a major outage. In Section 2, we introduce an extension to the failure model of [8] to not only consider the attacker and defender, but also the different types of nodes in the power grid (i.e. power generation vs. power consumers). In Section 3, we explore the computational complexity of finding deterministic best-response strategies for the attacker and defender under several different scenarios depending on the relative number of resources each player has and whether the opponent has a deterministic or mixed strategy. Here we found that, in general, these problems are NP-hard, though we do identify some tractable cases. In Section 4, we explore heuristic algorithms for finding determinsitic "best responses" as well as minimax mixed strategies. We introduce a "high-load" strategy for defense (based on the observations of [8]), greedy heuristics for deterministic strategies, and a double-oracle approach based on [15] for finding a mixed strategy. In Section 5 we perform experiments on a real-world dataset of a power grid [20] and find that this game seems to favor the attacker as he benefits more from increased resources than the defender. Further, our experiments revealed that the minimax defense produces roughly the same expected payoff as an easy-to-compute deterministic load based (DLB) defense when played against a minimax attack strategy, though the load based defense does more poorly than minimax when faced with the attacker's best response to DLB. This is likely due to the presence of low-load yet high-payoff nodes, which we also found in our empirical analysis of the model. Finally, related work is discussed in Section 6.

2. TECHNICAL PRELIMINARIES

Consider a power-grid network modeled as an undirected graph $G = (V, E)$. Let $V_{src}, V_{ld} \subseteq V$ be source (producers of power) and load (consumers of power) on the network. We shall use the notation $\mathsf{disc}_{V_{ld},V_{src}}(G)$ to denote the number of nodes in V_{ld} which are not connected to any node in V_{src} in graph G. Let $\mathbf{G}$ be the set of all subgraphs of G. For a given node i, let $\mathcal{N}_G(i)$ be the set of nodes in $V_{src} - \{i\}$ that are closest to that node (based on path length in G). From this, we define edge load (similar to the idea of edge betweenness [25]).

Definition 2.1 (Edge Load). *Given edge $ij \in E$, the* ***edge load****, $load_G(ij)$ is defined as follows:*

$$load_G(ij) = \sum_{t \in V_{ld}} \sum_{s \in \mathcal{N}_G(t)} \frac{\sigma_G(s,t|ij)}{|\mathcal{N}_G(t)|\,\sigma_G(s,t)},$$

where $\sigma_G(s,t)$ is the number of shortest paths between $s, t \in V$ and $\sigma_G(s,t|ij)$ is the subset of these paths that pass through edge $ij \in E$.

Starting from initial network $G_0 = (V_0, E_0)$ we use c_{ij} to denote the capacity edge $ij \in E_0$. In a real-world setting, we would expect to have this information. However, in this paper, we use the following proxy (similar to [8]).

$$c_{ij}(G_0) = (1 + \alpha) load_{G_0}(ij)$$

where α is a non-negative real that specifies the excess capacity available on that line. We shall refer to α as the *capacity margin*. We assume that an edge $ij \in E$ fails in $G = (V, E)$, with $E \subset E_0$, if $load_G(ij) > c_{ij}(G_0)$. Once nodes (and adjacent edges) in V_0 are removed from G_0, this results in a change of shortest paths between sources and loads, hence more edges will potentially fail. This cascading power failure is modeled by a "failure" operator denoted with F (based on the failure model of [8] - though we note that our model is a new contribution due to the consideration of source and load nodes) that maps networks to networks. We define it as follows.

Definition 2.2 (Failure Operator). *The* ***failure operator****, $\mathsf{F} : \mathbf{G} \to \mathbf{G}$, is defined as follows:*

$$\mathsf{F}((V,E)) = (V, \{ij \in E | load_{(V,E)}(ij) \le c_{ij}(G_0)\})$$

Intuitively, one application of the failure operator removes all edges that have exceeded their maximum capacity. We can define multiple applications of this operator as follows:

$$\mathsf{F}^i(G) = \begin{cases} G & \text{if } i = 0 \\ \mathsf{F}(\mathsf{F}^{i-1}(G)) & \text{otherwise} \end{cases}$$

Clearly, there must exist a fixed point that is reached in no more than $|E| + 1$ applications of F. Hence, we shall use the following notation:

$$\mathsf{F}^*(G) = \mathsf{F}^i(G) \ s.t.\ \mathsf{F}^i(G) = \mathsf{F}^{i+1}(G)$$

We now consider two agents: an attacker and a defender. The attacker's strategy is to destroy nodes (and their adjacent edges) in an effort to cause a cascading failure that maximizes the number of load nodes (V_{ld}) that are disconnected from all source nodes (V_{src}). Meanwhile, the defender's strategy is to harden certain nodes such that the attacker is unable to destroy them - though these nodes can be taken offline as a result of the cascading failure[1]. The attacker can destroy k_a nodes while the defender can harden k_d nodes. Thus the strategy space of both the attacker and defender consists of all subsets $V_a, V_d \subseteq V$ of size $|V_a| \le k_a$ ($|V_d| \le k_d$ respectively). We denote these strategy spaces by ATK (DEF respectively), i.e., if we allow the attacker to consider all strategies of size k_a or less we have:

$$ATK = \{S \in 2^V : |S| \le k_a\}$$

We now have all of the components to define the payoff function.

Definition 2.3 (Payoff Function). *Given initial network $G = (V, E)$ with edge capacities $c_{ij}(G)$, attack (defend) strategy $V_a(V_d)$, the payoff function is defined by*

$$p_G(V_a, V_d) = \mathit{disc}_{V_{ld},V_{src}}(\mathsf{F}^*((V - (V_a - V_d), E)).$$

Now, in reality, the defender will have real-world limitations on the number of nodes (i.e. substations) he may harden. For instance, with regard to smart grid defense, applying the most up-to-date patches on all systems may not be realistic as it could potentially require system down-time - affecting customer service. Further, it would also likely not make sense for the defender to only harden certain nodes and ignore others. Hence, it is reasonable to consider a situation where the defender can only harden certain nodes against attack (and may do so probabilistically - i.e. applying hardware or software updates according to a schedule). Therefore, we study mixed strategies. Such strategies will be specified by probability distributions $\mathbf{Pr}_a, \mathbf{Pr}_d$ for the attacker and defender respectively. We shall denote the number of strategies assigned a non-zero probability as $|\mathbf{Pr}_a|, |\mathbf{Pr}_d|$. We can define expected payoff as follows.

Definition 2.4 (Expected Payoff). *Let $\boldsymbol{Pr}_a, \boldsymbol{Pr}_d$ be probability distributions over all subsets of V of sizes k_a (resp. k_d) or less. These probability distributions correspond to a mixed strategy for the attacker and defender respectively. Hence, given such probability distributions, the expected payoff can be computed as follows:*

$$\boldsymbol{ExP}(\boldsymbol{Pr}_a, \boldsymbol{Pr}_d) = \sum_{V_a \in 2^V} \boldsymbol{Pr}_a(V_a) \sum_{V_d \in 2^V} \boldsymbol{Pr}_d(V_d) p_G(V_a, V_d)$$

In this work our goal is to find the *minimax* strategy for the defender - that is the mixed strategy for the defender that minimizes the attacker's maximum expected payoff - as well as deterministic "best responses" for both players given the other's strategy.

3. COMPUTATIONAL COMPLEXITY

In this section, we analyze the computational complexity of determining the best response for each of the agents to a strategy of its opponent. First, we shall discuss the case for finding a deterministic strategy for the defender and attacker. Then we shall explore the computational complexity of finding a mixed strategy. We summarize our complexity results in Table 3.

[1] Note that this would likely be the case where the attack and defense occurs in cyber-space, while the cascade occurs in the physical world.

Opponent Strategy	Attacker	Defender
Mixed w. 1 resource	NP-Compl. Thm. 3	PTIME Prop. 3.2
Det. w. fewer resources	NP-Compl. Thm. 3	PTIME Prop. 3.1
Det. w. greater resources	NP-Compl. Thm. 3	NP-Compl. Thm. 1
Mixed w. fewer resources	NP-Compl. Thm. 3	NP-Compl. Thm. 2
Mixed w. greater resources	NP-Compl. Thm. 3	NP-Compl. Thm. 1

Table 1: Complexity Results for Finding a Deterministic Best Response

We frame the formal combinatorial problem of finding the best-response for the defender as follows:

Grid-Defend Deterministic Best Response (GD-DBR)
INPUT: Network $G = (V, E)$, attacker mixed strategy $\mathbf{Pr}_a$ (where each option is of size no greater than k_a), natural number k_d, real numbers X, α
OUTPUT: "Yes" if there exists a set $V_d \subseteq V$ s.t. $|V_d| \leq k_d$ and $\sum_{V_a \in ATK} \mathbf{Pr}_a(V_a) p_G(V_a, V_d) \leq X$ and "no" otherwise.

We shall study this case under several conditions. The first, and easiest case is when $\mathbf{Pr}_a = 1$ (the attacker uses a deterministic strategy) and $k_a \leq k_d$.

Proposition 3.1. *When $k_a \leq k_d$ and $|\boldsymbol{Pr}_a| = 1$ then GD-DBR is solvable in polynomial time.*

Proof. As the attacker plays only one strategy and the defender can defend at least as many nodes as are being attacked, the defender simply defends all the nodes in the attacker's strategy. □

However, even with $|\mathbf{Pr}_a| = 1$, the problem becomes NP-hard in the case where $k_a > k_d$.

Theorem 1. *When $k_a > k_d$ then GD-DBR is NP-complete, even when $|\boldsymbol{Pr}_a| = 1$ and X is an integer.*

Proof. Clearly, checking if a given deterministic defender strategy V_d meets the requirements of the "output" of GD-DBR can be completed in polynomial-time, providing membership in the class NP.

For NP-hardness consider the known NP-hard "set cover" problem [11] that takes as input a natural number k, set of elements $S = \{s_1, \ldots, s_n\}$, family of subsets of S, $H = \{h_1, \ldots, h_m\}$ and returns "yes" if there is a k-sized (or smaller) subset of H s.t. their union is equal to S. We can embed Set Cover into an instance of GD-DBR in polynomial time with the following embedding: set $k_a = |H|$, $k_d = k$, $X = 0$, $\alpha = |H| + |S|$, create $G = (V, E)$ as follows:

- For each $h \in H$ create a node v_h and for each $s \in S$ create node v_s
- If $s \in h$, create edge (v_h, v_s), for each $ij \in E$
- Set $V_{src} = \{v_h | h \in H\}$, $V_{ld} = \{v_s | s \in S\}$, $V_a = V - V_{ld}$

Suppose, by way of contradiction (BWOC), that there is a "yes" answer to Set Cover but a "no" answer to GD-DBR. Consider set H' a subset of H that is the certificate for Set Cover and the corresponding set $V' = \{v_h | h \in H'\}$ in the instance of GD-DBR. Suppose the defender utilizes this as a strategy. The attacker then effectively attacks the set $V - V_{ld} - V'$. Note that as the graph is bi-bipartite, this does not cause any cascading failure. By the construction, each load node must be connected to a source node, hence the number of offline load nodes is X. This gives us a contradiction.

Suppose, BWOC, that there is a "yes" answer to GD-DBR but a "no" answer to the corresponding instance of Set Cover. Let V' be the certificate for GD-DBR. We note that any element of $V_{ld} \cap V'$ in V' can be replaced by a neighboring node from V_{src} without changing the size of this set and that such a set would still allow for all load nodes to remain online, let V'' be this new set. Consider the set $\{h | v_h \in V''\}$. By the contra-positive of the claim, this cannot be a cover of all elements of S. However, this would also imply that there is some element $v_s \in V_{ld}$ that is not connected to V'' meaning that it fails (as the attacker successfully destroys all its neighbors). This means that the adversary has a payoff greater than 0 (which is what X was set to) – hence a contradiction. □

Hence, the presence of a more advantageous attacker is a source of complexity. The next question would be if the attacker's behavior, i.e. deterministic vs. non-deterministic, also affects the complexity of the problem, even if the defender has the advantage. First, let us examine the case where the attacker has a mixed strategy with $k_a = 1$.

Proposition 3.2. *When $k_a = 1$ then GD-DBR is solvable in polynomial time (w.r.t. $|\boldsymbol{Pr}_a|$), even when $|\boldsymbol{Pr}_a| \geq 0$.*

Proof. In this case, we can re-write the payoff function as $p_G(\{v\}, V_d) = 0$ if $v \in V_d$ and $p_G(\{v\}, V_d) = p_G(\{v\}, \emptyset)$ otherwise. Let $V' = \cup\{V_a \in ATK | \mathbf{Pr}_a(V_a) > 0\}$. Note that each element of V' is also a strategy the attacker plays with a non-zero probability (as the attacker only plays singletons). Hence, the expected payoff can be re-written as $\sum_{v \in V' - V_d} \mathbf{Pr}_a(\{v\}) p_G(\{v\}, \emptyset)$. Therefore, the best a defender can do is defend the top k_d nodes in V' where $\mathbf{Pr}_a(\{v\}) p_G(\{v\}, \emptyset)$ is the greatest - which can be easily computed in polynomial time and allows us to determine the answer to GD-DBR. □

However, if the defender is playing a mixed strategy with $k_a > 1$, then the problem again becomes NP-complete.

Theorem 2. *When $|\boldsymbol{Pr}_a| > 1$ and $k_a > 1$, GD-DBR is NP-complete, even when $k_d > k_a$ and X is an integer.*

Proof. NP-completeness mirrors that of Theorem 1. For NP-hardness, we again consider a reduction from set-cover (defined in the proof of Theorem 1. The embedding can again be performed in polynomial time as follows: set $k_a = \max_{s \in S} |\{h | s \in h\}|$, set $k_d = k$, $X = 0$, $\alpha = |H| + |S|$, create $G = (V, E)$, V_{src}, and V_{ld} as per the construction in Theorem 1. We then set up the mixed strategy as follows: for each $s \in S$, let $V_a^s = \{h | s \in h\}$ and $\mathbf{Pr}_a(V_a^s) = 1/|S|$.

Suppose, BWOC, that there is a "yes" answer to set cover and a "no" answer to the instance of GD-DBR. Consider set cover solution H^* and set $V_d = \{v_h | h \in H^*\}$. Note that V_d meets the cardinality requirement. Note that by the construction, a source node becomes disconnected only if all of the load nodes connected to it are attacked, hence there is some node in the set V_{ld} that is totally disconnected under at least one attacker strategy - let v_s be this node.

However, as set H^* covers S, then regardless of the attacker strategy, there is always some node v_h that is connected and never attacked (giving the attacker a payoff of zero) - hence a contradiction.

Suppose, BWOC, that there is a "yes" answer to GD-DBR and a "no" answer to the instance of set cover. Consider GD-DBR solution V'. We note that any element of $V_{ld} \cap V'$ in V' can be replaced by a neighboring node from V_{src} without changing the size of this set and that such a set would still allow for all load nodes to remain online, let V'' be this new set. Consider the set $H^* = \{h | v_h \in V''\}$. Note that $|H^*| \leq k$. By the contra-positive, there must be at least one element of S not covered by H^*. Let node v_s be a node associated with uncovered element s. As GD-DBR returned "yes" then there is no attacker strategy where v_s becomes disconnected from some node in V_{src}. As attack strategy V_a^s includes all nodes that are connected to v_s, then at least one of these nodes must be included in V''. Therefore, for every node $v_s \in V_{ld}$ there is some node $v_h \in V_{ld} \cap V''$ that is connected to it, which means, by the construction, that H^* must cover all elements of S - a contradiction. □

We now frame the formal problem for finding a deterministic best-response for the attacker below.

Grid-Attack Deterministic Best Response (GA-DBR)
INPUT: Network $G = (V, E)$, defender mixed strategy $\mathbf{Pr}_d$ (where each option is of size no greater than k_d), natural number k_a, real numbers X, α
OUTPUT: "Yes" if there exists a set $V_a \subseteq V$ s.t. $|V_a| \leq k_a$ and $\sum_{V_d \in DEF} \mathbf{Pr}_d(V_d) p_G(V_a, V_d) \geq X$ and "no" otherwise.

In the case of $k_a = 1$, this problem is solvable in polynomial time: simply consider each $v \in V$. The attacker computes $\sum_{V_d \in DEF} \mathbf{Pr}_d(V_d) p_G(\{v\}, V_d)$ until one is found that causes the payoff to exceed or be equal to X. However, for strategies of larger size, the problem becomes NP-hard, regardless of the size of the defender strategy.

FACT 3.1. *When $k_a = 1$, GA-DBR is solvable in polynomial time (w.r.t. $|\mathbf{Pr}_d|$).*

THEOREM 3. *GA-DBR is NP-complete.*

PROOF. Clearly, a certificate consisting of a set $V_a \subseteq V$ can be verified in polynomial time, giving us membership in NP. For NP-hardness consider the known NP-hard "vertex cover" problem [11] that takes as input a graph $G' = (V', E')$ (with no self-loops) and natural number k and returns "yes" iff there is a set of k or fewer vertices that are adjacent to each edge in E. We can embed vertex cover into an instance of GD-DBR in polynomial time with the following embedding: set $k_a = k$, $k_d = 0$, $V_d = \emptyset$, $X = |V'|$, $\alpha = |E|$, $G = G'$, and $V_{src} = V_{ld} = V'$.

Suppose, BWOC, the above problem instance provides a "yes" answer to the vertex cover problem but a "no" answer to GA-DBR. Let V'' be a vertex cover of size k or less for G'. Consider the corresponding set of vertices in G (we shall call this V^*). Note that $|V^*| \leq k_a$. As an attacker attacking V^* disconnects those nodes from the network, all edges adjacent to V^* fail. As V^* is a vertex cover for G, this means that there are no edges in the graph once V^* is removed. Hence, no load node is connected to any source node - giving the attacker a payoff of at least X – hence a contradiction.

Suppose, BWOC, the above problem instance provides a "yes" answer to GA-DBR but a "no" answer to the vertex cover problem. Let V_a be the set of nodes the attacker attacks in GA-DBR. As $\alpha = |E|$ and as $V_{src} = V$, nodes only fail in a cascade if they are either targeted by the attacker or become totally disconnected. Further, as $X = |V|$, all nodes in G are either in V_a or disconnected - meaning that V_a must be a vertex cover of size k_a or less. As $k_a = k$ we have a contradiction. □

Due to the use of covering problems for the complexity results in Theorems 1, 2, and 3, it may seem reasonable to frame the problem as a sub- or super- modularity optimization where the objective function is monotonic. However, here we show (unfortunately) that these properties do not hold for either player. First, we shall make statements regarding the monotonicity of the payoff function.

PROPOSITION 3.3. *Iff $\forall V_d^*$, $V_a \subseteq V_a'$: $p_G(V_a, V_d^*) \leq p_G(V_a', V_d^*)$ then $\forall V_a^*$, $V_d \subseteq V_d'$: $p_G(V_a^*, V_d) \geq p_G(V_a^*, V_d')$.*

The idea of *submodularity* can be thought of as "diminishing returns." Given a set of elements S and a function $f : 2^S \rightarrow \Re^+$, we say a f is submodular if for any sets $S_1 \subseteq S_2$ and element $s \notin S_2$, we have the following relationship:

$$f(S_1 \cup \{s\}) - F(S_1) \geq f(S_2 \cup \{s\}) - F(S_2)$$

A complementary idea of supermodularity is also often studied - in this case the inequality is reversed. Unfortunately, when we fix the strategy for the defender, the attacker strategy is neither submodular nor supermodular - making the dynamics of this model significantly different from others (i.e. [24]). Let consider strategies V_a, V_d where V_a causes some load node $v \notin (V_a \cup V_d) \cap V_{ld}$ to disconnect and any node the strategy $\{v\}$ causes to disconnect will also become disconnected with strategy V_a (such a case is easy to contrive, particularly with a bi-partite network). Therefore, we get the following relationship:

$$p_G(V_a \cup \{v\}, V_d) - p_G(V_a, V_d) < p_G(\{v\}, V_d) - p_G(\emptyset, V_d)$$

This arises from the fact that the left-hand side of the above equation becomes zero and the right hand side of the equation is equal to $p_G(\{v\}, V_d)$ which must be at least one. Now consider another example. Suppose we have a simple V-shaped network of three nodes. The angle of the V is a load node, while the other two nodes are source nodes. With $\alpha = 1$, the load node receives power if at least one of the source nodes is connected to it. However, it does not require both. Let V_a be a strategy consisting of one source node and v be the other source node, and V_d consist of the load node. From this, we have the following relationship:

$$p_G(V_a \cup \{v\}, V_d) - p_G(V_a, V_d) > p_G(\{v\}, V_d) - p_G(\emptyset, V_d)$$

In this case, the right-hand side becomes zero while the left hand side becomes one. This leads us to the following fact:

FACT 3.2. *When V_d is fixed, p_G is neither submodular nor supermodular.*

Now let us consider when we fix the attacker's strategy. If the payoff is submodular when the attacker's strategy is fixed, then we have the following for $V_d \subseteq V_d'$ and $v \notin V_d'$ if the payoff subtracted from the number of nodes is submodular:

$$p_G(V_a, V_d' \cup \{v\}) - p_G(V_a, V_d') \geq p_G(V_a, V_d \cup \{v\}) - p_G(V_a, V_d)$$

This is equivalent to the following:

$$p_G(V_a - (V_d' \cup \{v\}), \emptyset) - p_G(V_a - V_d', \emptyset) \geq$$
$$p_G(V_a - (V_d \cup \{v\}), \emptyset) - p_G(V_a - V_d, \emptyset)$$

Now let $V_a' = V_a - (V_d' \cup \{v\})$ and $V_a'' = V_a' \cup (V_d' - V_d)$. Clearly $V_a'' \supseteq V_a'$ and $v \notin V_a''$. Now we get the following:

$$\begin{aligned} p_G(V_a', \emptyset) - p_G(V_a' \cup \{v\}, \emptyset) &\geq p_G(V_a'', \emptyset) - p_G(V_a'' \cup \{v\}, \emptyset) \\ p_G(V_a' \cup \{v\}, \emptyset) - p_G(V_a', \emptyset) &\leq p_G(V_a'' \cup \{v\}, \emptyset) - p_G(V_a'', \emptyset) \end{aligned}$$

Hence, submodualrity of the payoff function when the attacker's strategy is fixed would give us supermodualrity of the payoff function when the defender's strategy is fixed at the empty set. However, this clearly violates Fact 3.2 and gives rise to the following:

FACT 3.3. *When V_a is fixed, p_G is neither submodular nor supermodular.*

4. ALGORITHMS

In this section, we present heuristic algorithms for finding the deterministic best response of each player as the results of the previous section generally preclude a polynomial time algorithm for an exact solution. We first introduce a version of a "high load" strategy for the defender based on the ideas of [8]. Then we introduce a greedy heuristic for each player. This is followed by our approach for finding mixed strategies based on the double-oracle algorithm of [15].

Hi-Load Node Approach. In [8], the authors study "high load" nodes: nodes through which the greatest number of shortest paths pass. They show that attacks on these nodes tend to initiate cascading failures – suggesting that they should be a priority for defense. We formalize the definition of nodal load in our framework (essentially an extended definition of node betweenness [25]) by extending our function $load_G$ for nodes as follows.

DEFINITION 4.1 (NODAL LOAD). *For a given node, the **nodal load** is defined as the sum of the fraction of shortest paths for each pair that pass through that node. Formally:*

$$load_G(i) = \sum_{s \in V_{src}, t \in V_{ld}} \frac{\sigma_G(s,t|i)}{\sigma_G(s,t)},$$

where $\sigma_G(s,t|i)$ is the number of shortest paths between $s,t \in V$ that pass through node i.

Hence, we shall refer to the *Deterministic Load-Based* or DLB strategy for the defender as one in which he deterministically protects the k_d nodes with the greatest load. We note that this is not necessarily a "best response" but the intuition is that defense will occur at nodes that are perceived to be critical to the adversary. This intuition is similar to that of the "most vital arc" idea seen in other failure model games [2, 21].

Greedy Heuristics for Finding Deterministic Strategies. Here we present a simple greedy heuristic to find the defender's best-response (GREEDY_DEFENDER_RESP). The analogous heuristic for the attacker is not shown due to space constraints, but we shall refer to it as GREEDY_ATTACKER_RESP. We note that while we do not make general approximation guarantees (due to the results in Section 3), we note that by Proposition 3.3, that nodes added in step 18 will always cause an increase in payoff to the defender (and in the analogous greedy approach for the attacker, this holds true as well). Further, by Proposition 3.2, when $k_a = 1$, we can be sure that GREEDY_DEFENDER_RESP returns an exact solution, even when the attacker has a mixed strategy. Unfortunately, by Theorem 3, the same cannot be said if the greedy heuristic is used for the attacker's best response.

Algorithm 1 GREEDY_DEFENDER_RESP

```
Require: Mixed strategy $\mathbf{Pr}_a$, Natural number $k_d$
Ensure: Set of nodes $V_d$

 1: $V_d = \emptyset$
 2: Let $ATK$ be the set of strategies associated with $\mathbf{Pr}_a$
 3: Set $flag$ = True, $p^* = -\infty$
 4: while $|V_d| \leq k_d$ and $flag$ and $p^* < 0$ do
 5:   $p^* = -\sum_{V_a \in ATK} \mathbf{Pr}_d(V_a) p_G(V_a, V_d)$
 6:   $curBest = null$, $curBestScore = 0$, $haveValidScore$ = False
 7:   for $i \in V - V_d$ do
 8:     $curScore = p^* - \sum_{V_a \in ATK} \mathbf{Pr}_d(V_a) p_G(V_a, V_d \cup \{i\})$
 9:     if $curScore \geq curBestScore$ then
10:       $curBest = i$
11:       $curBestScore = curScore$
12:       $haveValidScore$ = True
13:     end if
14:   end for
15:   if $haveValidScore$ = False then
16:     $flag$ = False
17:   else
18:     $V_d = V_d \cup \{curBest\}$
19:   end if
20: end while
21: return $V_d$.
```

Finding Mixed Strategies. If the attacker uses a mixed strategy that consists of uniformly attacking elements of $\{S \subset V_{ld} : |S| = k_a\}$ then the best any pure defender strategy can do is defending $V_d \subset V_{ld}$. The attacker's strategy implies that any node in V_{ld} is attacked with probability $\frac{k_a}{|V_{ld}|}$. Each of the $|V_{ld}| - k_a$ remaining nodes in V_{ld} is then disconnected with probability $\frac{k_a}{|V_{ld}|}$, i.e., $x \geq k_a(1 - \frac{k_d}{|V_{ld}|})$. Clearly due to the cascading the value of the game will probably be higher, illustrating the disadvantage the defender has in this game. To determine both player's optimal strategies and the value of the game we resort to an algorithmic approach. We find the defender's optimal strategy with the following linear program. We can find minimax strategy for the defender with the following linear program. It simply assigns a probability to each of the defenders strategies in a manner that minimizes the maximum payoff for the adversary. As a consequence, the solution to the following linear program, DEF_LP can provide the mixed minimax strategy for the defender. An analogous linear program, ATK_LP (not shown), which mirrors DEF_LP, will provide that result for the attacker.

Definition 4.2 (DEF_LP).

$$\min p^* \quad (1)$$

$$\textit{subj.to} \quad p^* \geq \sum_{V_d \in DEF} X_{V_d} p_G(V_a, V_d) \quad \forall V_a \in ATK \quad (2)$$

$$1 = \sum_{V_d \in DEF} X_{V_d} \quad (3)$$

$$X_{V_d} \in [0,1] \quad \forall V_d \in DEF \quad (4)$$

Note that the above linear program requires one variable for each of the defender's strategies and one constraint for each of the attacker's strategies. However, as there are a combinatorial number of strategies, even writing down such a linear program is not practical except for very small problem instances. To address this issue of intractability, we employ the double-oracle framework for zero-sum games introduced in [15] and has been applied in more recent work as well [5, 12]. We present the algorithm DOUBLE_ORACLE as follows:

Algorithm 2 DOUBLE_ORACLE

```
Require: Network $G = (V, E)$, natural number $maxIters$
Ensure: Mixed defender strategy $\mathbf{Pr}_d$
Initialize $numIters = 0$, $flag =$ True
Initialize the sets of strategies $ATK, DEF$ to both be $\{\emptyset\}$
while $flag$ and $numIters \leq maxIters$ do
    Create $\mathbf{Pr}_a, \mathbf{Pr}_d$ based on the solutions to ATK_LP and DEF_LP respectively.
    IF $numIters < maxIters$ THEN let $V_a$ be the attacker's best response to $\mathbf{Pr}_d$ and $V_d$ be the defender's best response to $\mathbf{Pr}_a$
    IF $V_a \in ATK$ and $V_d \in DEF$ THEN $flag =$ False ELSE $ATK = ATK \cup \{V_a\}$, $DEF = DEF \cup \{V_d\}$
    $numIters += 1$
end while
return $\mathbf{Pr}_a$.
```

The intuition behind the above algorithm is that it iteratively creates mixed strategies for both the attacker and defender based on a solution to a linear program over the sets of current possible strategies for both players (ATK, DEF). This is followed by finding (for each player) the best deterministic response to it's opponent's strategy. If these new strategies are both already in the set of possible strategies for the respective players, the algorithm terminates. Otherwise, they are added to ATK, DEF respectively. We note that by Theorem 1 of [15] that the above algorithm will guarantee an exact solution if $maxIters$ is set to the number of possible strategies. In practice, [15] demonstrates that the algorithm converges much faster.

In DOUBLE_ORACLE, the finding the solutions to DEF_LP, ATK_LP will be tractable provided that the algorithm converges in a polynomial number of steps (either through convergence or after the specified $maxIters$). However, as we have shown, computing the best responses is usually computationally difficult. Although, we note in the case where $k_a = 1$, that by Proposition 3.2 and Fact 3.1, the double oracle algorithm will return an optimal solution, even if greedy approximations are used for the oracles (provided it runs until convergence).

5. EXPERIMENTAL EVALUATION

All experiments were run on a computer equipped with an Intel X5677 Xeon Processor operating at 3.46 GHz with a 12 MB Cache and 288 GB of physical memory. The machine was running Red Hat Enterprise Linux version 6.1. Only one core was used for experiments. All algorithms were coded using Python 2.7 and leveraged the NetworkX library[2] as well as the PuLP library for linear programming[3]. All statistics presented in this section were calculated using the R statistics software.

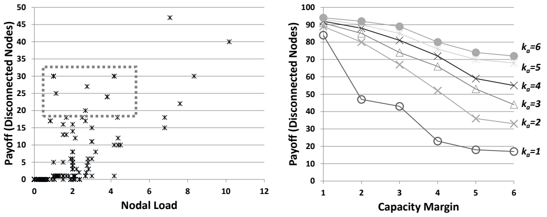

Figure 1: Left: Nodal load vs. payoff (note hi-payoff, low-load nodes in the dashed box), Right: Capacity margin (α) vs. payoff

In our experiments, we utilized a dataset of an Italian 380 kV power transmission grid [20]. This power grid network consisted of 310 nodes of which 113 were source, 96 were load, and the remainder were transmission nodes. The nodes were connected with 361 edges representing the power lines.

In our initial experiments, we examined the properties of the model when no defense is employed. In Figure 1 (left) we show results concerning nodal load vs. the payoff achieved by the adversary if that node is attacked (and no others). Interestingly, we noticed a significant number of nodes with low nodal load yet high-payoff if attacked (see nodes in dashed box). This may suggest that the DLB strategy may be insufficient in some cases. Later we see how DLB fails to provide adequate in a defense against the attacker best response to DLB. This is likely due to these hi-payoff, low-load nodes. In Figure 1 (right) we examine α (capacity margin) vs. attacker payoff for various settings of k_a (using the GREEDY_ATTACKER_RESP heuristic). Here we found that, in general, payoff decreases linearly with capacity margin ($R^2 \geq 0.84$ for each trial).

Next, we examined the relative performance of the minimax (mixed) defense strategy and the DLB strategy under different resource constraints and against the minimax (mixed) attack strategy as well as the attacker's (deterministic) greedy response to the DLB defense. In these experiments, we considered the case where both players have equal resources, the attacker has one resource (which by Proposition 3.2 and Fact 3.1 we are guaranteed an optimal solution), and the defender has one resource. These results are displayed in Figure 2. In these trials we set the capacity margin $\alpha = 0.5$, meaning that all edges had an excess capacity of 50%. We did not use the $maxIters$ parameter of the DOUBLE_ORACLE algorithm, but instead allowed it to run until convergence.

With regard to the comparison between DLB and minimax defense, both performed comparably against the minimax attack strategy. In fact, an analysis of variance (ANOVA) indicated little variance between the two when faced with the minimax attacker ($p \geq 0.74$ for these trials). Yet, a defender known to be playing a single strategy would likely not face an attacker who plays the minimax strategy, but rather the

[2] http://networkx.lanl.gov/

[3] http://pythonhosted.org/PuLP/

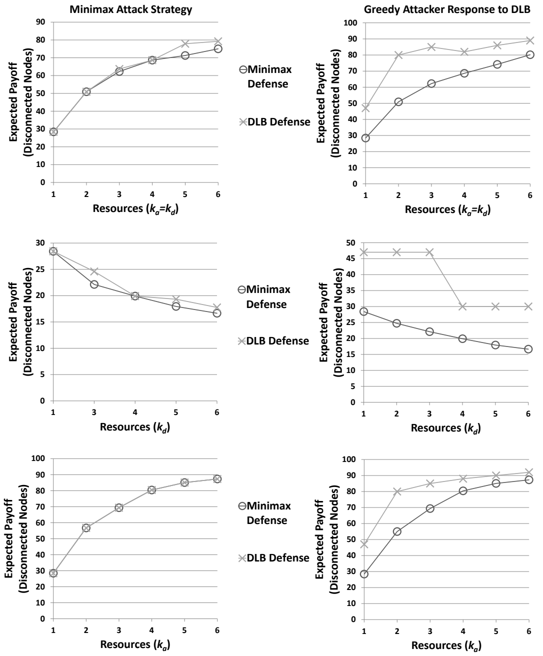

Figure 2: Minimax and DLB defense strategies vs. minimax attack strategy (left) and the attacker's greedy best response to DLB (right). Examined are the cases where $k_a = k_d$ (top), $k_a = 1$, k_d varies (middle) and $k_d = 1$, k_a varies (bottom).

best response to the DLB. In this case, DLB play resulted in significantly greater payoff to the attacker than the defender ($p \leq 0.29$ for these trials, the DLB defense results in 15.6 more disconnected nodes on average). This failure of the DLB strategy to perform well against a deterministic attacker best response is likely due to the presence of low-load yet high-payoff nodes as shown in Figure 1.

We also noticed that an increase in resources seems to favor the attacker more than the defender. When both players played their respective minimax strategy, the expected payoff for the attacker increased monotonically with the cardinality of the strategies. Further, when $k_d = 1$ and k_a was greater, the attacker's payoff tripled when his resources increased from 1 to 6. However, when $k_a = 1$ and k_d was greater, the defender's payoff only increased by a factor of 1.7. Hence, the attacker can cause more damage than the defender can mitigate with the same amount of extra resources. We suspect that this is likely because a defended node can still fail during a cascade - which would likely be the case if the attack and defense operations are restricted to cyber-space, where physical system failure may still be possible as the result of a cascade initiated by virtual means.

We also examined the run-time of our approach, as displayed in Figure 3 (left). Though run-time did seem to scale linearly with strategy size ($R^2 = 0.90 \pm 0.2$ for each experiment), it appears that run-time will in general prohibit the study of larger strategies or networks (our longest experiment ran for 12 days). In examining the iterations of the DOUBLE_ORACLE algorithm, Figure 3 (left), we find that run-time of an iteration of the algorithm progressively increases (note that this figure is showing the run-time for each iteration, not a cumulative time). This increase is likely the combined result of the growing linear program and the growing size of the mixed strategies considered by the greedy approximation sub-routines. We are currently exploring reliable methods to limit the number of iterations while maintaining defender payoff.

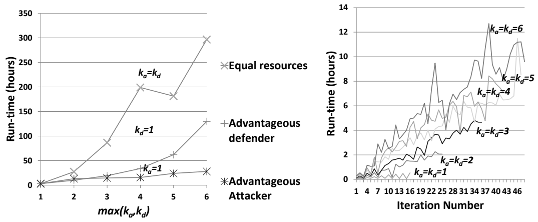

Figure 3: Strategy size vs. run-time in hours (left) and the run-time of each iteration for the experiments where $k_a = k_d$

6. RELATED WORK

Network security has received much attention from the research community in the past two decades. Recent incidents have shown that due to their internet connectedness such networks can come under cyber attack, causing severe problems[4]. See [26] for a discussion of cyber-security issues relevant to smart grid grids.

The utilization of game theory in designing defense solutions seems ubiquitous. For instance [13] model the interaction between a DDoS attacker and the network administrator while [14] considers a game theoretic formulation for intrusion detection. Other formulations consist include stochastic games [17], signaling games [19], allocation games [4] and repeated games [3]. Game theory is also being used in monitoring and decision making in smart grids, see for instance [9] or the survey by Fadlullah et al. [10]. However to date no game theoretic approach has been given for the specific problem where the attacker explicitly sets of a cascading power failure to maximize the damage to the defender.

Cascading failure models applied to power grid infrastructure have been studied in the past [7, 8, 16]. The model of [8] introduces the idea of edge failure based on excessive loads. The goal of the research presented in these papers was to illustrate properties of the cascade, rather than explore strategies for attack and defense as this work does. There has been work on attack and defense of a power-grid network under the DC power-flow mode [2, 21, 20, 6]. However, the DC power flow model is not designed to model the more rapid cascading failures (i.e. the 2003 cascading failure in the eastern United States [1]).

The application of game theory to security situations was made popular by [18] where it used for airport security patrol scheduling. Since then, other applications have emerged including port protection [23], finding weapons caches [22], and security checkpoint placement [12]. One that bears similarity to this work is [24] - studying games for controlling contagions on a network. However, as previously discussed, that model operates under very different dynamics.

[4] http://www.wired.com/threatlevel/2009/10/smartgrid/

7. CONCLUSION

In this paper, we explored complexity, algorithmic, and implementation issues in a two-player security game where the attacker/defender look to create/mitigate cascading failure on a power grid. Future work includes an examination of scalability issues (larger networks and strategies), adding uncertainty to the model, and the consideration of more real-world information about the power grid network (i.e. actual line capacities, etc.) in order to create a richer model.

8. ACKNOWLEDGMENTS

We would like to thank D. Alderson for his input on related work and V. Rosato for providing us the power grid dataset. Some of the authors are supported by ARO project 2GDATXR042. The opinions in this paper are those of the authors and do not necessarily reflect the opinions of the funders, the U.S. Military Academy, or the U.S. Army.

9. REFERENCES

[1] Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations. *U.S.-Canada Power System Outage Task Force*, April 2004.

[2] D. L. Alderson, G. G. Brown, M. W. Carlyle, and L. Anthony Cox. Sometimes there is no ”most-vital” arc: Assessing and improving the operational resilience of systems. *Military Operations Research*, 18(1):21–37, 2013-03-01T00:00:00.

[3] T. Alpcan and T. Basar. A game theoretic analysis of intrusion detection in access control systems. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 2, pages 1568–1573 Vol.2, 2004.

[4] M. Bloem, T. Alpcan, and T. Başar. Intrusion Response as a Resource Allocation Problem. *Decision and Control, 2006 45th IEEE Conference on*, pages 6283–6288, Dec. 2006.

[5] B. Bosanský, C. Kiekintveld, V. Lisý, J. Cermak, and M. Pechoucek. Double-oracle algorithm for computing an exact nash equilibrium in zero-sum extensive-form games. In *AAMAS*, pages 335–342, 2013.

[6] G. Brown, M. Carlyle, J. Salmeron, and K. Wood. Defending critical infrastructure. *Interfaces*, 36(6):530–544, Nov. 2006.

[7] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin. Catastrophic cascade of failures in interdependent networks. *Nature*, 464(7291):1025–1028, Apr. 2010.

[8] P. Crucitti, V. Latora, and M. Marchiori. Model for cascading failures in complex networks. *Phys. Rev. E*, 69(4):45104, 2004.

[9] M. Esmalifalak, G. Shi, Z. Han, and L. Song. Bad data injection attack and defense in electricity market using game theory study. *IEEE Trans. Smart Grid*, 4(1):160–169, 2013.

[10] Z. Fadlullah, Y. Nozaki, A. Takeuchi, and N. Kato. A survey of game theoretic approaches in smart grid. In *Wireless Communications and Signal Processing (WCSP), 2011 International Conference on*, pages 1–4, 2011.

[11] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[12] M. Jain, V. Conitzer, and M. Tambe. Security scheduling for real-world networks. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2013.

[13] P. Liu, W. Zang, and M. Yu. Incentive-based modeling and inference of attacker intent, objectives, and strategies. *ACM Trans. Inf. Syst. Secur.*, 8(1):78–118, Feb. 2005.

[14] Y. Liu, C. Comaniciu, and H. Man. A bayesian game approach for intrusion detection in wireless ad hoc networks. In *Proceeding from the 2006 workshop on Game theory for communications and networks*, GameNets '06, New York, NY, USA, 2006. ACM.

[15] H. B. McMahan, G. J. Gordon, and A. Blum. Planning in the presence of cost functions controlled by an adversary. In T. Fawcett and N. Mishra, editors, *ICML*, pages 536–543. AAAI Press, 2003.

[16] A. E. Motter and Y. C. Lai. Cascade-based attacks on complex networks. *Phys. Rev. E*, 66(6), Dec. 2002.

[17] K. C. Nguyen, T. Alpcan, and T. Basar. Security games with incomplete information. In *ICC*, pages 1–6. IEEE, 2009.

[18] P. Paruchuri, J. P. Pearce, J. Marecki, M. Tambe, F. Ordonez, and S. Kraus. Playing games for security: an efficient exact algorithm for solving bayesian stackelberg games. In *AAMAS*, pages 895–902, Richland, SC, 2008.

[19] A. Patcha and J.-M. Park. A game theoretic approach to modeling intrusion detection in mobile ad hoc networks. In *Information Assurance Workshop, 2004. Proc. from the Fifth Annual IEEE SMC*, pages 280–284, 2004.

[20] V. Rosato, L. Issacharoff, F. Tiriticco, S. Meloni, S. D. Porcellinis, and R. Setola. Modelling interdependent infrastructures using interacting dynamical models. *IJCIS*, 4(1/2):63–79, 2008.

[21] J. Salmeron, K. Wood, and R. Baldick. Analysis of electric grid security under terrorist threat. *Power Systems, IEEE Transactions on*, 19(2):905–912, May 2004.

[22] P. Shakarian, J. P. Dickerson, and V. S. Subrahmanian. Adversarial geospatial abduction problems. *ACM Trans. Intell. Syst. Technol.*, 3(2):34:1–34:35, Feb. 2012.

[23] E. Shieh, B. An, R. Yang, M. Tambe, C. Baldwin, J. DiRenzo, B. Maule, and G. Meyer. Protect: a deployed game theoretic system to protect the ports of the united states. In *AAMAS*, pages 13–20, Richland, SC, 2012.

[24] J. Tsai, T. H. Nguyen, and M. Tambe. Security games for controlling contagion. In J. Hoffmann and B. Selman, editors, *AAAI*. AAAI Press, 2012.

[25] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Number 8 in Structural analysis in the social sciences. Cambridge University Press, 1 edition, 1994.

[26] D. Wei, Y. Lu, M. Jafari, P. Skare, and K. Rohde. Protecting smart grid automation systems against cyberattacks. *Smart Grid, IEEE Transactions on*, 2(4):782–795, 2011.